

INICIAR SESIÓN

NUESTROS PLANES

TODOS LOS  
CURSOS

FORMACIONES

CURSOS

PARA  
EMPRESAS

## ARTÍCULOS DE TECNOLOGÍA

# Nueva exigencia de autenticación por token en Git ¿qué es y qué debo hacer?



Camila Alves

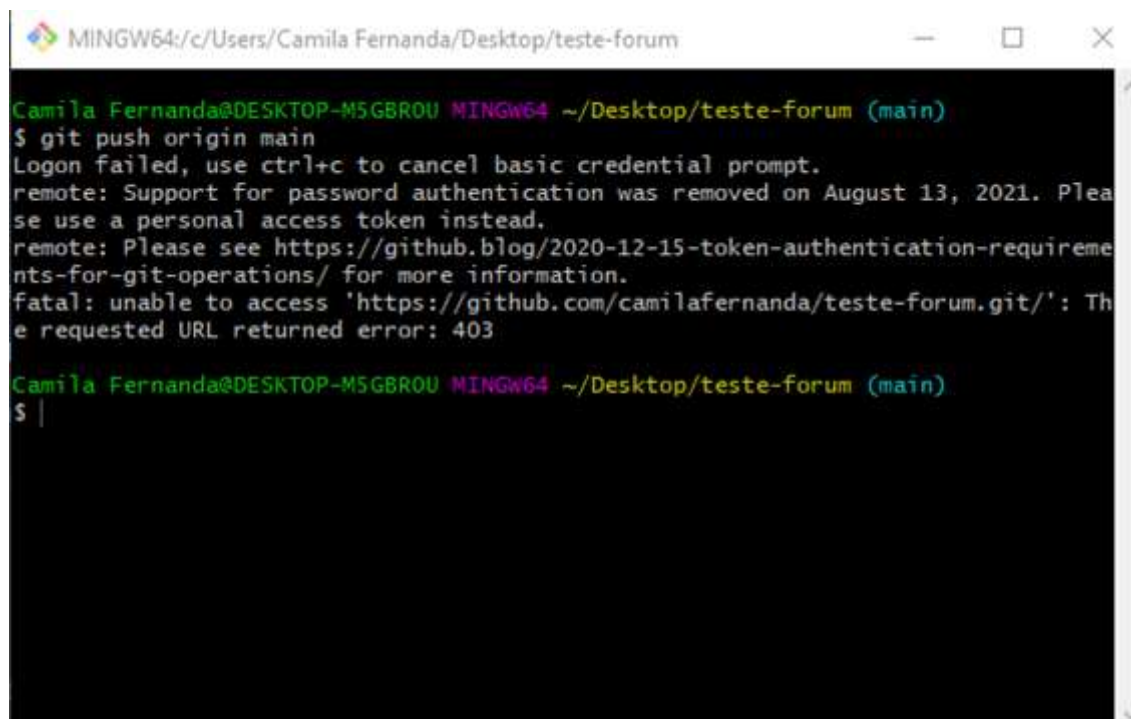
24 de Marzo



¿Últimamente has encontrado este mensaje al ejecutar algunos comandos en git?

```
Logon failed, use ctrl+c to cancel basic credential prompt.  
remote: Support for password authentication was removed on August 13, 2021.  
Please use a personal access token instead.  
remote: Please see https://github.blog/2020-12-15-token-authentication-require  
fatal: unable to access '<repositório Git>': The request URL returned error: 4
```



A screenshot of a terminal window titled 'MINGW64/c:/Users/Camila Fernanda/Desktop/teste-forum'. The prompt is 'Camila Fernanda@DESKTOP-M5GBR0U MINGW64 ~/Desktop/teste-forum (main)'. The user enters '\$ git push origin main'. The terminal output shows: 'Logon failed, use ctrl+c to cancel basic credential prompt.', 'remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.', 'remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.', and 'fatal: unable to access 'https://github.com/camilafernanda/teste-forum.git/': The requested URL returned error: 403'. The prompt returns to '\$ |'.

Esto sucedió, porque Github [anunció](#) un nuevo requisito de autenticación basado en token para Git. A partir del día 13 de agosto de 2021, la contraseña de la cuenta de Github ya no es aceptada para autenticar operaciones en Git.

Este cambio se debe por **motivos de seguridad**, ya que los tokens ofrecen algunos beneficios, como la posibilidad de guardar las informaciones del usuario en el token y no en el servidor, así como el hecho de poder configurarse para tener los permisos que desees y tener un periodo de expiración. Es decir, deja de funcionar después de un determinado tiempo, como si fuera una contraseña única y desechable.

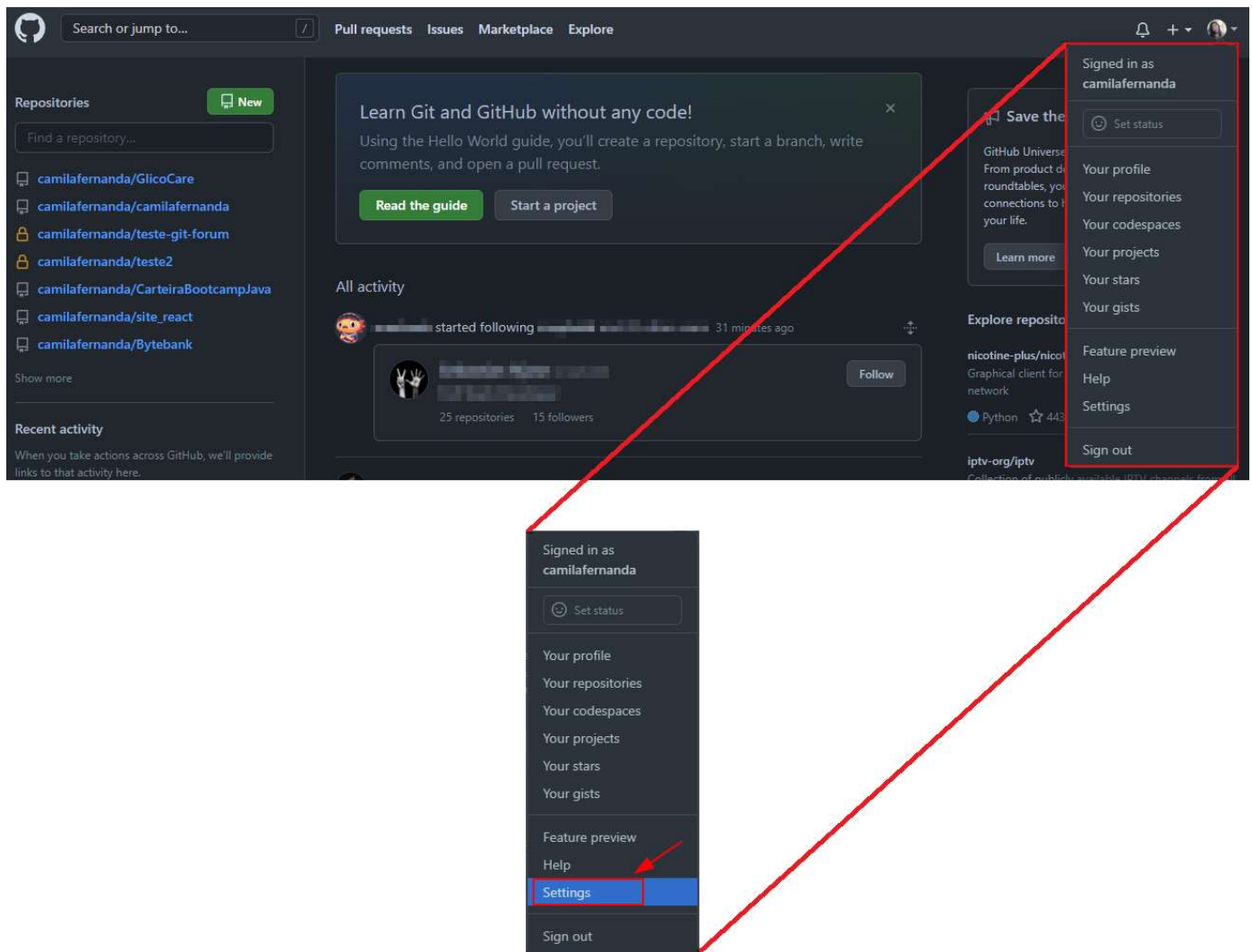
De esta forma, si otra persona tiene acceso a tu token, se restringirá únicamente los permisos que hayas configurado previamente. Entonces, no tendrás el riesgo de perder tu cuenta de Github, ya que puedes descartar el token perdido y generar otro nuevamente.

## Entonces ¿qué es necesario hacer para continuar usando las operaciones Git?

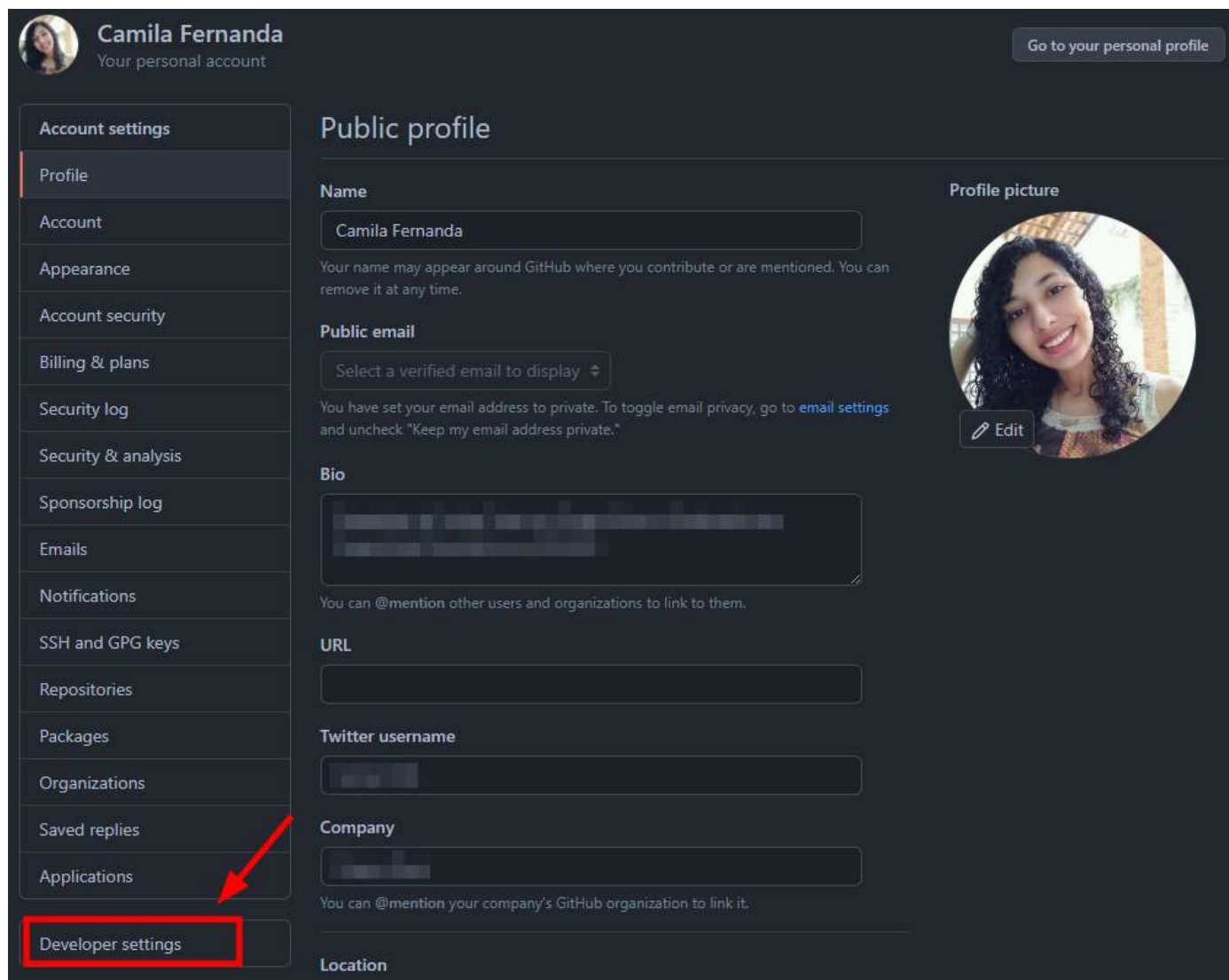
Github recomienda empezar a utilizar un token de acceso personal, pero si lo deseas, puedes optar por usar una clave SSH.

De esta manera, sigue un paso a paso de como generar tu token personal:

1) En la página de [Github](#), entra en tu cuenta. 2) Ya en la página principal, selecciona tu usuario en el rincón superior derecho y haz clic en la opción **Settings**:



3) En la página que abrirá, haz clic en la opción **Developer settings**, 24ª opción de la lista a la izquierda.



Camila Fernanda  
Your personal account

Go to your personal profile

Account settings

- Profile
- Account
- Appearance
- Account security
- Billing & plans
- Security log
- Security & analysis
- Sponsorship log
- Emails
- Notifications
- SSH and GPG keys
- Repositories
- Packages
- Organizations
- Saved replies
- Applications
- Developer settings**

## Public profile

**Name**

Camila Fernanda

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

**Public email**

Select a verified email to display

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

**Bio**

You can @mention other users and organizations to link to them.

**URL**

**Twitter username**

**Company**

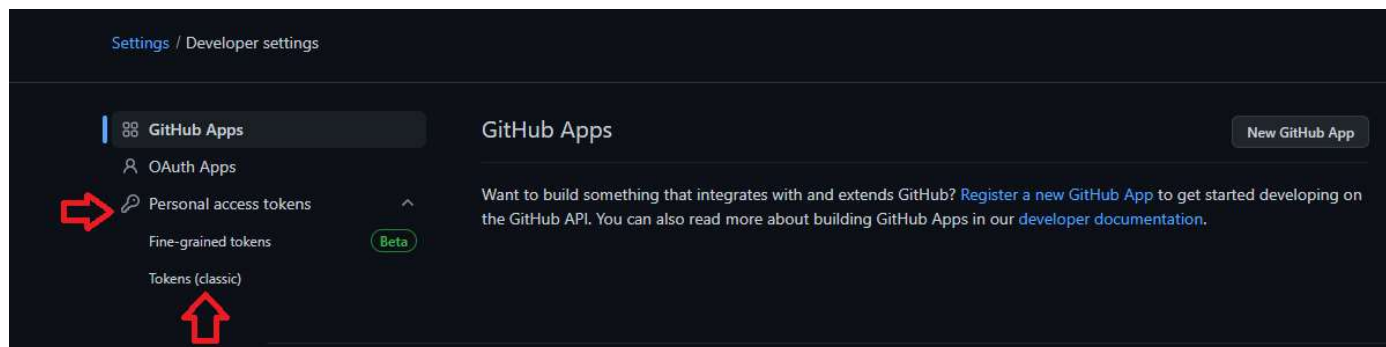
You can @mention your company's GitHub organization to link it.

**Location**

**Profile picture**

Edit

4) Selecciona la opción **Personal access tokens**, 3ª opción de la lista a la izquierda. Luego selecciona **Tokens(classic)**, 2ª opción en la lista que se abre luego abajo.



Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

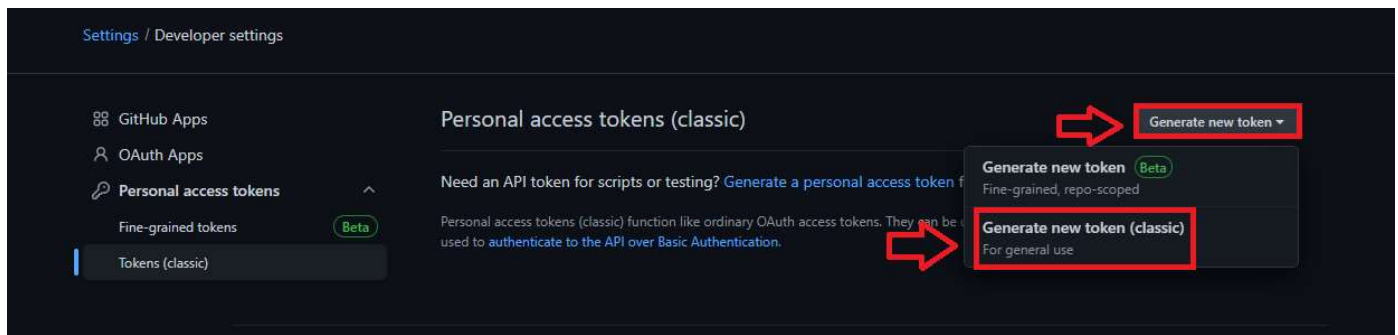
Tokens (classic)

GitHub Apps

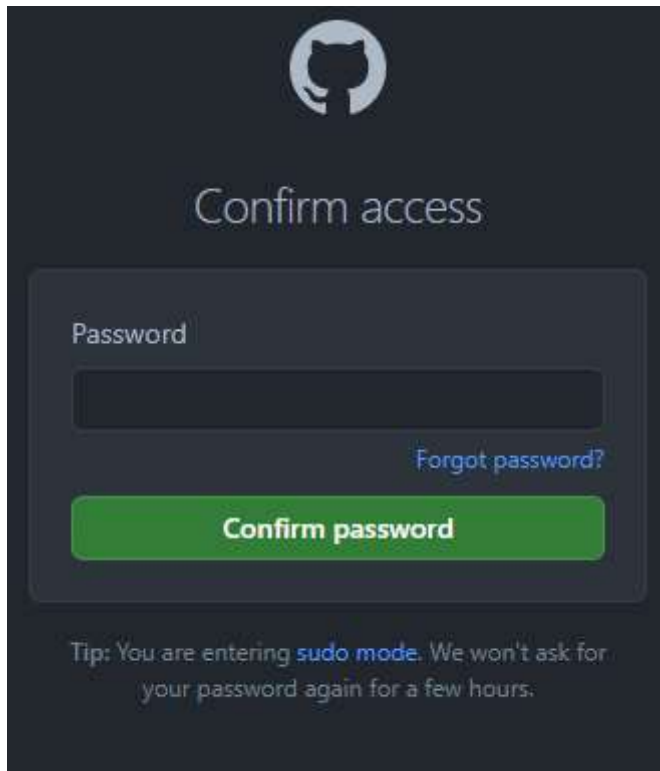
New GitHub App

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

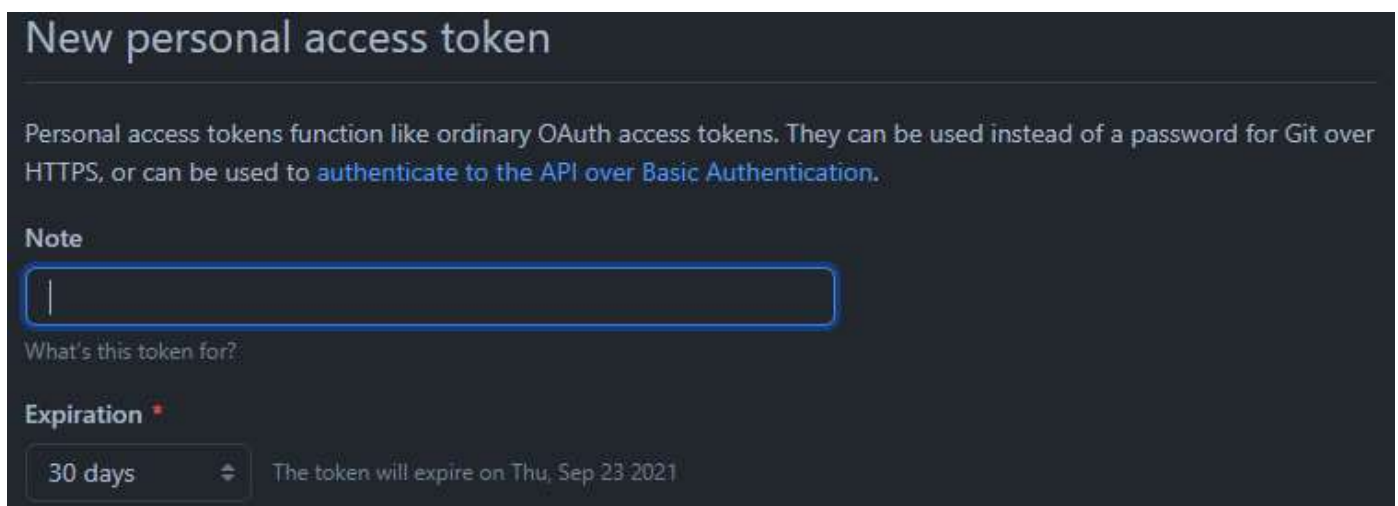
5) Haz clic en la opción **Generate new token** en la parte superior izquierda y luego en **Generate new token (classic)** que es la 2ª opción en la nueva lista que se abre.



6) Confirma con tu contraseña de Github.



7) Di un nombre para tu token y define el tiempo de expiración.



8) Marca los permisos que deseas conceder al token. Un detalle, para usar el token para acceder a repositorios, selecciona la opción **repo**.

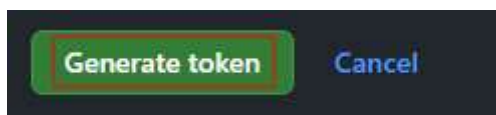


**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry
<input type="checkbox"/> <b>admin:org</b>	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> <b>admin:public_key</b>	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> <b>admin:repo_hook</b>	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> <b>admin:org_hook</b>	Full control of organization hooks
<input type="checkbox"/> <b>gist</b>	Create gists
<input type="checkbox"/> <b>notifications</b>	Access notifications
<input type="checkbox"/> <b>user</b>	Update ALL user data
<input type="checkbox"/> read:user	Read ALL user profile data
<input type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users
<input type="checkbox"/> <b>delete_repo</b>	Delete repositories

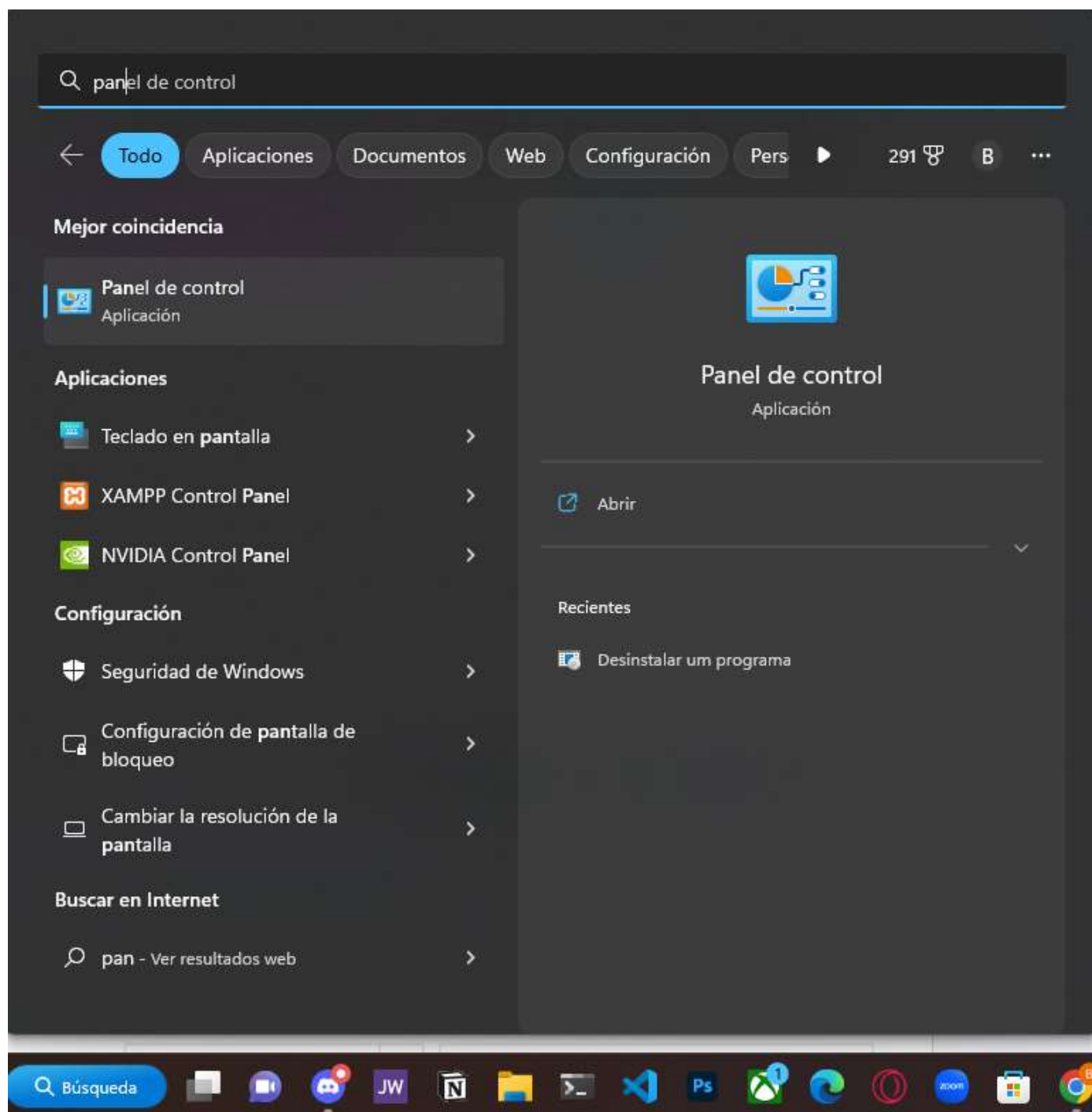
9) Por último, al seleccionar todas las opciones elegidas, haz clic en **Generate token** al final de la pantalla.



Listo, aparecerá un código para que puedas copiar y pegar en el Administrador de Credenciales de tu máquina. Este proceso dependerá de tu sistema operacional.

## Como editar credenciales en Windows

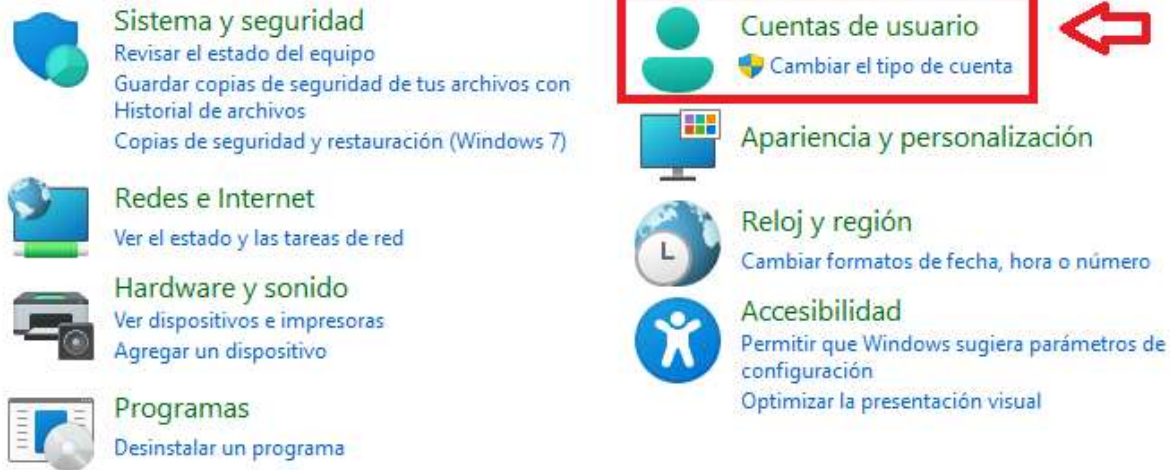
Después que generas el token, cópialo y accede al **Panel de Control**. Lo puedes encontrar digitando en la barra de búsqueda.



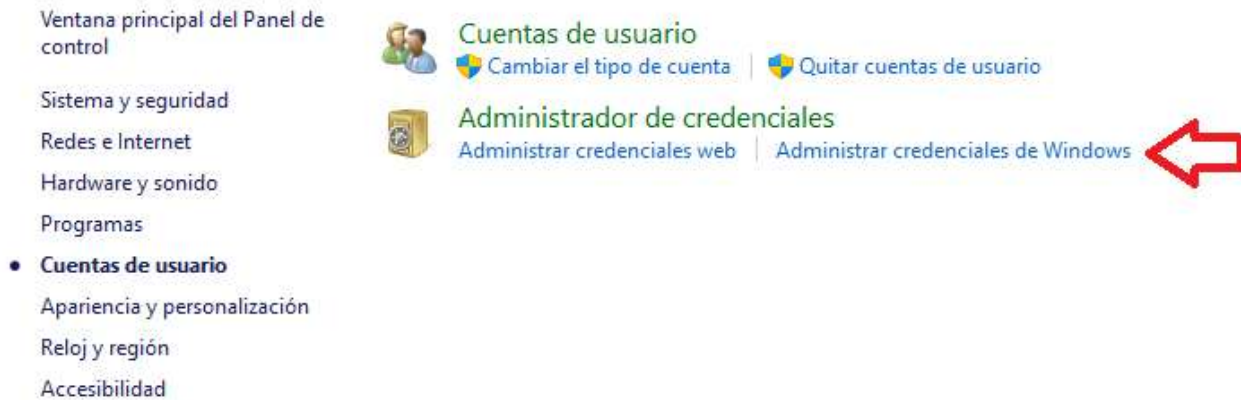
En seguida, accede a **Cuentas de Usuario**:

Ajustar la configuración del equipo

Ver por: Categoría ▾



Haz clic en **Administrar Credenciales de Windows**:



En **Credenciales de Windows** busca la sección de **Credenciales Genéricas** o **github.com** selecciónalo y ponlo a **editar**;



Ventana principal del Panel de control

[Copia de seguridad de credenciales](#) [Restaurar credenciales](#)

### Credenciales de Windows

### Agregar una credencial de Windows

No hay credenciales de Windows.

Credenciales basadas en certificados

[Agregar una credencial basada en certificado](#)

No hay certificados.

### Credenciales genéricas

[Agregar una credencial genérica](#)

git:https://github.com

Modificado: 25/8/2022

Copia el token generado en la primera etapa y pega en la caja de texto de la **Contraseña**.

[Editar credencial genérica](#)

Asegúrese de que el nombre de usuario y la contraseña que escriba se pueden usar para acceder a la ubicación.

Dirección de red o Internet: `git:https://github.com`

Nombre de usuario:

darkmoonsk

Contraseña:

●●●●●●●●

Guardar

Cancelar

En caso de que no encuentres el github.com, puedes seleccionar la opción **Agregar una credencial genérica**

[Copia de seguridad de credenciales](#) [Restaurar credenciales](#)

Credenciales de Windows

[Agregar una credencial de Windows](#)

No hay credenciales de Windows.

Credenciales basadas en certificados

[Agregar una credencial basada en certificado](#)

No hay certificados.

Credenciales genéricas



[Agregar una credencial genérica](#)



Y llenar los campos:

- Dirección de red o Internet: <https://github.com/>
- Nombre de usuario: El nombre de usuario que tienes en Github
- Contraseña: El token personal generado anteriormente

#### Escriba la dirección del sitio web y la información de sus credenciales

Asegúrese de que el nombre de usuario y la contraseña que escriba se pueden usar para acceder a la ubicación.

Dirección de red o Internet:

Nombre de usuario:

Contraseña:

[Aceptar](#)

[Cancelar](#)

¡Listo! Luego podrás realizar operaciones de Git nuevamente sin solicitudes de inicio de sesión hasta la fecha de expiración del token generado.

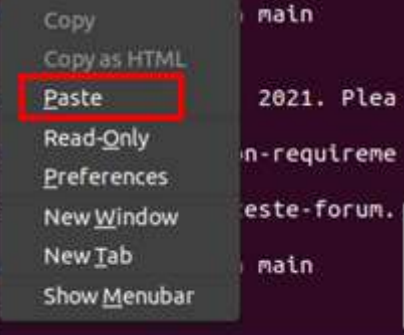
## Cómo editar una credencial de Git en Linux

En Linux, este proceso solo se puede realizar en la terminal. Al abrir un repositorio de Git en él y ejecutar comandos como `git pull`, `git push` o `git clone`, Git solicitará usuario y contraseña. Si introduces tu contraseña de GitHub como se hacía anteriormente, recibirás el mensaje citado al principio:

```
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git pull origin main
Username for 'https://github.com': camilafernanda
Password for 'https://camilafernanda@github.com':
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.
fatal: Authentication failed for 'https://github.com/camilafernanda/teste-forum.git/'
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$
```

De esta manera, simplemente tienes que ejecutar el comando de nuevo y en lugar de la contraseña, pegar tu token generado:

```
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git pull origin main
Username for 'https://github.com': camilafernanda
Password for 'https://camilafernanda@github.com':
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.
fatal: Authentication failed for 'https://github.com/camilafernanda/teste-forum.git/'
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$
```



Así, el comando se ejecutará con éxito:

```
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git pull origin main
Username for 'https://github.com': camilafernanda
Password for 'https://camilafernanda@github.com':
From https://github.com/camilafernanda/teste-forum
 * branch                main                -> FETCH_HEAD
Already up to date.
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$
```

Sin embargo, aunque funcione, no es viable tener que pegar el token cada vez que se ejecutan estos comandos, ¿verdad? Para ello, simplemente escribe el siguiente comando para que la última contraseña utilizada se guarde:

```
git config --global credential.helper cache
```

De esta forma, en el próximo comando que se ejecute, Git solicitará la autenticación nuevamente, pero en los siguientes comandos, Git no pedirá tu nombre de usuario y contraseña de nuevo.

```
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git config --global credential.helper cache
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git push origin main
Username for 'https://github.com': camilafernanda
Password for 'https://camilafernanda@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.85 KiB | 1.85 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/camilafernanda/teste-forum.git
 5843a25..9e3eab1  main -> main
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git push origin main
Everything up-to-date
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$
```

Y al final, en caso de que quieras quitar el token de la configuración, haz lo siguiente:

```
git config --global --unset credential.helper
```

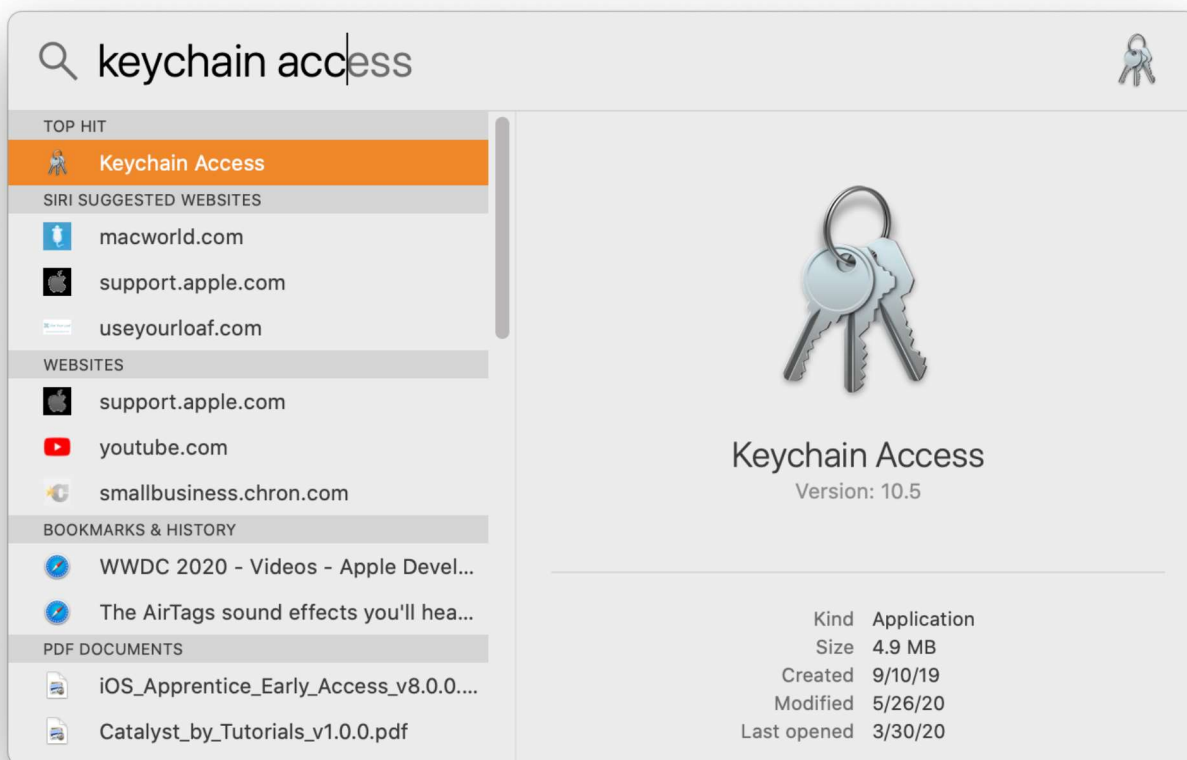
De esta manera, volverás a tener las solicitudes de autenticación nuevamente:

```
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git config --global --unset credential.helper
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git push origin main
Username for 'https://github.com': camilafernanda
Password for 'https://camilafernanda@github.com':
```

## Cómo editar las credenciales en MacOS.

Para configurar el token en Mac, el proceso es similar al de Windows. Accede a Keychain Access, para encontrarlo puedes hacer clic en el icono de búsqueda en la esquina derecha de la barra de menú:





Fuente: Imagen retirada de la [Documentación de Github](#)

A continuación, busca por **github.com**, haz clic en **internet password**, edita y pega el token personal generado.

Si quieres eliminarlo, en la terminal escribe:

```
git credential-osxkeychain erase
```

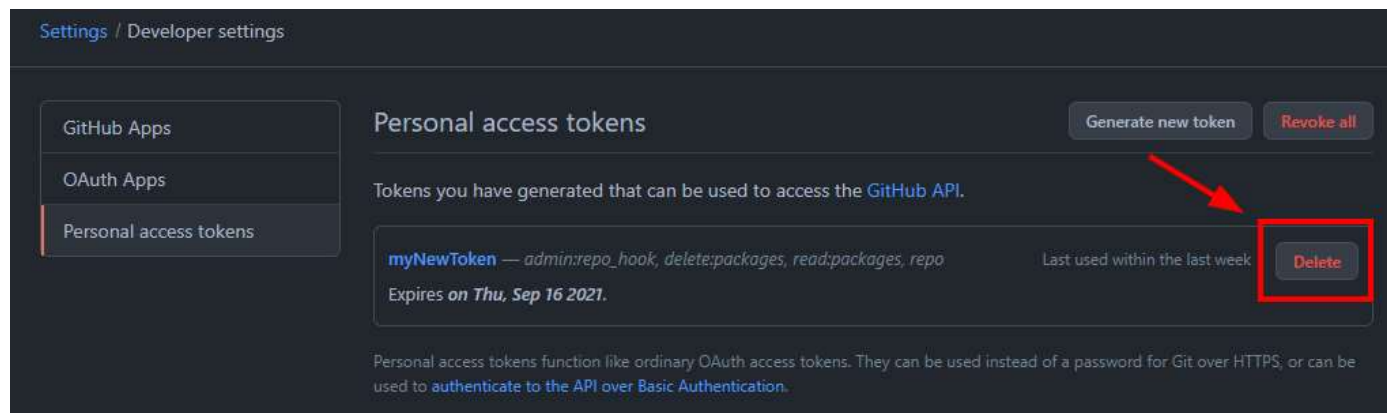
## Conclusión

En resumen, este cambio en el uso de tokens para autenticación es muy interesante, ya que puede ofrecer un mayor control y seguridad. Puede realizar todo el proceso mencionado anteriormente tantas veces como sea necesario, así como configurar los permisos y el tiempo de expiración de cada token generado de la manera que consideres mejor.

En caso de fallas de seguridad, por ejemplo, la persona estará limitada a los permisos otorgados al token y su cuenta de GitHub seguirá a salvo. Además, si esto sucede, elimina



tu token yendo a GitHub, luego a Settings, Developer settings y Personal access tokens, el mismo procedimiento para generar el token, pero haz clic en Delete:



Así estarás seguro nuevamente.



Camila Fernanda Alves

Soy casi ingeniera de la Universidad Federal de Uberlândia e instructora aquí en Alura. He trabajado como pasante en el Scuba Team y he realizado algunas tutorías de Física, Circuitos Eléctricos e incluso Procesamiento de Señales. Actualmente, estoy comenzando a producir contenido relacionado con DevOps y también estoy aquí para sumergirme en la tecnología con ustedes y, siempre que sea posible, tratar de ayudarlos a encontrar las respuestas que necesitan en los foros. En mi tiempo libre, me gusta jugar juegos de Nintendo y Valorant.

**Este artículo fue adecuado para Alura Latam por: [Bruno Souza](#)**

ARTÍCULOS DE TECNOLOGÍA

**En Alura encontrarás variados cursos sobre . ¡Comienza ahora!**

**SEMESTRAL**

**US\$49,90**

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

**ANUAL**

**US\$79,90**

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 12 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

Acceso a todos  
los cursos

Estudia las 24 horas,  
dónde y cuándo quieras

Nuevos cursos  
cada semana

## NAVEGACIÓN

PLANES

INSTRUCTORES

BLOG

POLÍTICA DE PRIVACIDAD

TÉRMINOS DE USO

SOBRE NOSOTROS

PREGUNTAS FRECUENTES

## ¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

## BLOG

PROGRAMACIÓN

FRONT END

DATA SCIENCE

INNOVACIÓN Y GESTIÓN

DEVOPS

AOVS Sistemas de Informática S.A

CNPJ 05.555.382/0001-33

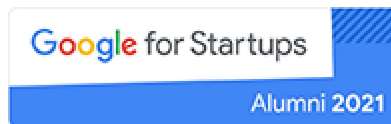
## SÍGUENOS EN NUESTRAS REDES SOCIALES



## ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

## CURSOS

Cursos de Programación



Lógica de Programación | Java

### **Cursos de Front End**

HTML y CSS | JavaScript | React

### **Cursos de Data Science**

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

### **Cursos de DevOps**

Docker | Linux

### **Cursos de Innovación y Gestión**

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics |

Liderazgo y Gestión de Equipos | Startups y Emprendimiento